Our Ref.: 493-35-3

U. S. PATENT APPLICATION

Inventor: Robert M. BEST

Invention: Networked Video Game Systems

Suggested class/subclass: 463-41

Graybeal Jackson Haley LLP
Attorneys at Law
155 108th Avenue NE, Suite 350
Bellevue, WA 98004-5973
(425) 455-5575
Fax (425) 455-1046

SPECIFICATION

TITLE OF THE INVENTION

Networked Video Game Systems

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This is a continuation in part of application Ser. No. 09/928,294 filed Aug. 10, 2001, which is a continuation in part of application Ser. No. 09/853,487 filed May 10, 2001, the entire content of which is hereby incorporated by reference herein.

FIELD OF THE INVENTION

[0002] This invention relates generally to electronic game systems and more particularly to networked multiple-player role playing games.

BACKGROUND OF THE INVENTION.

[0003] Networked video games are an established art as represented by USPTO patent Class/Subclass 463-41, for example US patent 6,579,184 (Tanskanen). It is known for networked client terminals to exchange status messages indicating object movement as described in US patents 6,437,777 (Kamachi et al.) and 6,570,563 (Honda). It is also known for personal computers to exchange messages through an Internet server providing Instant Messaging (IM) as described in US patent 6,539,421 (Appelman). It is also known for maps and other graphics to be displayed on an LCD in a portable game system linked to a video game console as described in US patent 6,500,070 (Tomizawa).

SUMMARY OF THE INVENTION

[0004] It is a primary objective of the present invention to provide multi-player games for players who operate separate video game systems that communicate over a network. In the preferred embodiment, networked video game consoles read data from software disks (with same title) and generate the same or related simulated 3D game worlds on different TV screens. Each player's friend uses a similar disk and may be miles away or next door. Using conventional control units, each player moves his player character in different parts of the simulated 3D world. Each console generates status data records providing information on the status of each game. These status records are transmitted to a common messaging server that transmits a copy of each data record to other client game systems for each small group of friends who agree in advance to play one common multi-player video game on their respective systems. As the game progresses, each player controls his own first player character, but is also made aware of what other player characters are doing, even if the other player characters are not close enough in the simulated game world to be visible on a TV screen view of the first player character.

[0005] Each game system receives frequent status data records from other game systems through the common messaging server and uses the status data to update variables for each character such as the location of the character in the simulated 3D world, the direction and velocity of movement of each character, the orientation of each character, collisions of characters and other objects, acquisitions of weapons, tools, etc. and which ones were acquired by each character, and energy and emotional attributes of player characters and non-player characters including player characters controlled by other players on the network. Status of objects and terrain in the simulated game world are also shared between

game systems, such as guns fired, points scored, portals entered, locked/unlocked doors and other obstacles and events encountered, fuel in a fuel tank, location of resources, etc.

[0006] Each player may view any of the other player characters and objects from any point of view in the simulated world. A player does not have to search for other characters, unless that is part of the game, because each game system provides an automatic display of the other characters on an LCD screen in a portable game system linked to a player's video game console. Each players can explore, discover, and view other parts of the game world and send the coordinates of discovered objects to other players in the network so that they can view and use the discovered objects. For example one player may discover a simulated storehouse of tools or materials that other players need.

[0007] There is no need for detailed pictures to be transmitted to other game systems because each game system uses software disks that provide the same programs to each game system which can automatically regenerate pictures of any selected character and other objects in the simulated game worlds with the same location, direction of movement, orientation, and other variables in the other game systems based on the status data in the transmitted status records.

ADVANTAGES

[0008] By using pre-existing instant messaging services or the telephone system to distribute frequent game status data among thousands of small groups of video game systems, video game companies need not set up their own expensive networks of game servers and Internet service providers. An added benefit is that thousands of small groups of game players can use the same messaging service to contact other players who are interested in playing a specific game title and agree on a time to begin playing. The same messaging service would then transmit hidden status messages to the game systems operated by those same players.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Fig. 1 is a perspective view of three human game players operating two video game consoles that are connected through the Internet to an instant messaging server.

[0010] Fig. 2 is a block diagram of an embodiment in which two video game consoles are connected through the Internet to an instant messaging server.

[0011] Fig. 3 is a perspective view of a game player controlling one video game console connected through the Internet to the console shown in Fig. 4.

[0012] Fig. 4 is a perspective view of a game player controlling one video game console connected through the Internet to the console shown in Fig. 3.

[0013] Fig. 5 is a perspective view of two game players who view both their own player characters and their friend's player characters on different display devices.

[0014] Fig. 6 is a perspective view of two game players who view their own player characters on TV screens while a map shows both player characters as symbols.

[0015] Fig. 7 is a perspective view of two game players controlling the same video game console that is connected through the Internet to other game consoles.

[0016] Fig. 8 is a perspective view and block diagram of the TV screen and portable game system illustrated in Fig. 7 and showing a virtual camera viewing objects that also appear as symbols on an LCD screen.

[0017] Fig. 9 is a block diagram of an embodiment in which two video game consoles are connected through personal computers and the Internet to an instant messaging server.

[0018] Fig. 10 is a block diagram of an embodiment in which two video game consoles are connected through short-range RF transceivers.

[0019] Fig. 11 is a block diagram of an embodiment in which two video game consoles are connected through a telephone network with no messaging server.

[0020] Fig. 12 is a flow chart illustrating processes performed by each video game system.

[0021] Fig. 13 are typical record formats of status records that may be transmitted to and received from an instant messaging server or direct through the Internet.

[0022] Fig. 14 are typical record formats of initializing records that may be transmitted to and received from an instant messaging server or direct through the Internet.

[0023] Fig. 15 is a perspective view of a game player controlling a video game console linked to two portable game systems operated as auxiliary display devices.

[0024] Fig. 16 is a perspective view of two game players controlling video game consoles connected through the Internet and using auxiliary display units.

[0025] Fig. 17 illustrates a player-selected sequence of pictures displayed on an LCD in a portable game system.

[0026] Fig. 18 is a detailed block diagram of a video game console system linked to a portable game system and to the Internet.

[0027] Fig. 19 is a time sequence example of two communicating game systems using semaphores to synchronize the two systems during delays in message transmission.

[0028] Fig. 20 is a memory map of the digital programs and data stored on a computer-readable data storage disk product.

[0029] Fig. 21 is a typical memory map of programs and data stored in RAM in each video game console.

[0030] Fig. 22 is a 3D cartesian graph illustrating 3D points of view of a virtual camera viewing a smulated object from different viewing angles.

[0031] Fig. 23 is a perspective view of a video game console linked to a portable game system that receives a memory cartridge having touch-tone button switches.

[0032] Fig. 24 is a block diagram illustrating downloading of game programs from a video game system to a portable game system.

Fig. 25 is a block diagram illustrating data flow between systems.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENT

[0033] Fig. 1 illustrates a preferred embodiment of a multi-player game system that has at least two video game consoles 42 sending video signals to corresponding television devices 11 for display as motion pictures. Each video game system 42 is manually controlled by at least one portable game system 44 and 47 being operated as a controller or controllers, in this example. Each game player 10 and 12 controls a different player character in the same multi-player game, typically a role playing game (RPG) shared by a small group of friends.

[0034] All of the video game systems 42 used by each group of friends execute the same game software during a game session, in this example. The same group of friends may use other game titles during different sessions. Other groups of friends may be using different game softare titles simultaneously in separate sessions.

[0035] Each of the video game systems 42 transmit status data records 78 (see Fig. 13) through the Internet or other communications network 121 to a messaging server 120 which forwards the status data records 78 to other video game systems 42 in each session.

[0036] Messaging server 120 operates like a prior-art instant messaging server that shares each natual language message with several friends in each "buddy list" as described in US Patent 6,539,421, except that in the Fig. 13 examples, status data records 78 contain mostly digital codes and numbers that need not be readable by game players and hence would typically not be displayed as instant messages. Status data records 78 are sent to the other video game systems 42 to be processed in each session to keep the systems synchronized so that each video game system 42 can display any character or object in a simulated game world from variable points of view at any time during each session.

[0037] Each player's television 11 displays his/her player character as usual, but other characters, objects, maps, scores, etc. are displayed on a portable game system 44, 47, in this example.

[0038] Fig. 2 illustrates an embodiment similar to the example shown in Fig. 1 and has additional manually operated handheld controllers 185 and 192 that can control corresponding video game systems 42 using control members 20 such as joystick, button switches, direction switches, and touch pads.

[0039] Each video game system in this example has a network interface 137 & 138 which may provide IP numbers and broadband access or a modem and a tone generator for dialing telephone numbers to access messaging server 120, typically through an Internet Service Provider (ISP).

[0040] Each status data record 78 sent to the other video game systems 42 in a session may pass through messaging server 120. Alternatively, messaging server 120 may determine the current network (IP) address of each video game system 42 when systems 42 are connected to server 120 and then transmit a set of IP address records (see Fig. 23) to each system 42 in a session. Then each system 42 may send and receive status data records 78 as packets through the Internet or other network using data transmission protocols such as TCP/IP, so that most of the data records 78 need not pass through server 120.

[0041] Fig. 3 illustrates part of a two-player network game system that is similar to that described above with reference to Fig. 2, except that in this Fig. 3 example, portable game system 44 is operated as an auxiliary display. Portable game system 44 is shown resting on table 187 and is linked to video game system 42 by cable 186 or wireless equivalent. Video game system 42 processes incoming status data records 13 and generates therefrom synchronizing data for transmission to portable game system 44 which causes processor 50 and coprocessor 301 (see Fig. 18) to generates image data representing motion pictures of variable 3D views of the simulated 3D game world(s). Video game system 42 may also transmit data to protable game system 44 for display as still pictures, maps, text, and other images on LCD screens 22.

[0042] Image data representing motion pictures of variable 3D views of the simulated 3D game world(s) may be generated as rendered polygons in both video game system 42 and in each portable game system 44 and 47 so that characters and other objects in the image data can be viewed from variable and rapidly changing points of view and angles selected by players. One player in a multi-player session may select a point of view that is different from the point of view selected by another player viewing the same character or object for display on two different LCD 22 screens on two different portable game systems 44 and 47 connected to two different video game systems 42. If by chance, both players select substantially the same point of view for viewing the same character, both LCD 22 screens will show substantially the same image at substantially the same time, even though no image or picture data was transmitted between the two systems. The images will be the same because the spacial coordinates and orientation of the characters are the same as a result of the synchronizing status data (Fig. 13) shared among all systems in a session.

[0043] Image data representing variable 3D views of the simulated 3D game world(s) generated by different video game system processors 86 and image co-processors 316 (Fig. 18) in multiple video game systems 42 for display on respective television screens 11, will typically not show the same view because each player will control a different player character and the television screen will usually display a different player character. However, if two player characters are controlled to move to a spatial location that is close to each other and both players by chance choose substantially the same point of view to view their respective characters, then the corresponding television screens 56 will show substantially the same scene that has both player characters at the same time.

[0044] CPU processor 86 and image co-processor 316 in each video game system 42 generate rendered polygons so that characters and other objects in the image data can be viewed on television screen 56 from variable and rapidly changing angles selected by players.

[0045] Manually operated handheld controller 185 generates control data in response to manipulation of control member 20 and other control members to controls video game system 42. Additional control data may be generated in portable game system 44 by manipulation of directional switch 15. By pressing a button switch 15 or direction switch 14 or joystick 20 (see Fig 18) on portable game system 44, a player can command system 42 to temporarily transfer control data from controller 185 through link 186 to portable system 44 to control images on LCD 22.

[0046] Alternatively, player 10 could retain control of his player character using controller 185 and use control members on portable system 44 to select points of view and locations

for display on LCD 22. Since this would be difficult for players with only two hands, this alternative is most suitable for multi-player games that accept control data from two players controlling the same video game system 42 as discribed below with reference to Fig. 7.

[0047] In Fig. 4, which is another part of the networked game described above with reference to Fig. 3, portable game system 47 is operated as an auxiliary display. Portable game system 47 is shown resting on table 187 and is linked to video game system 42 by cable 186 or wireless equivalent. Player 12, using controller 192, controls her player character 17 who is about to be eaten by a dinosaur, in this example, unless another player character, such as the man on the motorcycle controlled by player 10 in Fig. 3 can rescue character 17. Player 10 in Fig. 3 is watching the same scene on his portable system LCD 22 that player 12 sees on television screen 56 in Fig. 4, typically from different points of view selected by the different players 10 and 12.

[0048] In Fig. 5, players 10 and 12 can see each other's player characters on their respective portable game systems 44 and 47. For clarity only one controller is shown for each player in Fig. 5, but the embodiment described above with reference to Figures 3 and 4 is preferred. In Fig. 5 player 10 sees and controls his player character on the motorcycle on his television screen and also sees the dinosaur scene 33 on his portable game system 44 LCD 22. Likewise player 12 sees and controls her player character 17 on television screen 56 and also sees the motorcycle scene 34 on her portable game system 47 LCD 22.

[0049] The line of dashes down the middle of Fig. 5 is to emphasize that the game being played to the left of the dash line is distant from the game being played to the right of the dash line, as illustrated more clearly in Fig. 1.

[0050] By pressing a button on her portable game system 47 in Fig. 5 (or controller 192 in Fig. 4) player 12 may view on her system 47 LCD 22 the same scene 33 that player 10 is watching. This may reassure player 12 that player 10 is aware of the danger to character 17. This can be accomplished if the 3D point of view coordinates and 3D viewing angle for each player's LCD 22 display are transmitted to the video game systems 42 of other players in each session in status data records (not shown in Fig. 13).

[0051] Fig. 6 illustrates a map 33 being displayed on portable game system 44 LCD 22 that shows with geometric symbols an overview that covers both the motorcycle scene 34 and the dinosaur scene in map form. In this example, the rectangle in map 33 is the motorcycle and rider, the circle represents the dinosaur and the triangle represents character 17. All players in a session can display map 33 because the video game systems 42 in each session store data representing the same 3D world from which a map can be constructed or updated by processor 50 in portable game systems 44 and 47.

[0052] Fig. 7 illustrates another embodiment in which two players 10 and 12 are sitting in the same room and share a common television screen 56. Player 10 is not viewing a portable LCD in this example, because he is busy controlling his player character 18. Player 12 is not controlling the dinosaur scene, but is controlling the image being displayed on her portable game system 47 LCD 22 which in this example is the map 33 described above with reference to Fig. 6. As players 10 and 12 are coping with the dinosaur, other distant players are controlling their player characters in other parts of the same simulated 3D world as characters 17 and 18 and the dinosaur. Those other players can view on their portable LCD's the same scene that appears on television screen 56 in Fig. 7, because system 42 in Fig. 7 transmits status data to the other systems through Internet 121.

[0053] Fig. 8 illustrates a variable point of view as seen from a virtual "camera" 173 from the perspective of player controlled object 229. Television screen 56 displays the dinosaur 172 and character 18 which are represented as objects 172 and 18 at the top of Fig. 8 are are within the viewing angle of virtual camera 173. Objects 171 and object 228 are "off camera" and do not appear on television screen 56. On portable game system 47 LCD 22, all of the above objects appear as geometric symbols on a map including player object 229 which may be controlled by a third player (not shown) over the Internet.

[0054] Also displayed on the Fig. 8 LCD 22 map is cursor 49 which player 12 may control with direction switch 15 to position the cursor on an object or empty area of the map to select another point of view for display on LCD 22. Cursor 49 may also be used to select an object to control or otherwise make use of.

[0055] Fig. 9 illustrates another embodiment with makes uses of personal computers 137 and 138 to process data from connected keyboards on which to key IP numbers, phone numbers, account numbers, and passwords, and to provide software and access to an Internet Service Provider (ISP) and to send the account numbers and passwords to logon to a multi-player session. This may be used with video game systems 42 that lack a keyboard or network hardware or software. The connection between each video game system 42 may be with wires such as USB cables or a wireless connection 249 between transceivers 250 and 251.

[0056] Fig. 10 illustrates another embodiment which links two video game systems 42 with a short range radio connection for two game players who live only a short distance apart.

[0057] Fig. 11 illustrates another embodiment which links two video game systems 42 through modems 137 and 138 through a local dialup telephone system. This would be limited to two or three players who live in the same city. Each video game system 42 would have software for sharing status data records 78 with the other system 42.

[0058] Fig. 12 is a flowchart that illustrates examples of some of the processing performed by CPU processor 86 in each video game system 42. When a data connection is established through the Internet or other network between two or more video game systems 42, process 60 in each system 42 sends a record similar to records 78 in Fig. 13 to messaging server 120 to identify the player and the player character or object to be controlled by that player. Server 120 then sends a data record to each video game system 42 that identified the session number and game id and a common time/date to eliminate any time zone and clock synchronization problems. Each video game system 42 sends confimation data records back to the server which checks that all systems are synchronized.

[0059] After conflicts are corrected and this handshake process 60 is completed, process 60 initializes a "self status" table of data that specifies the initial spatial coordinates and orientation of each player character, and other initial data. Process 60 sends this initial data as status data records 78 to other systems.

[0060] Process 62 then updates this self status table if a player has moved his character.

[0061] Process 65 then checks for any incoming status data records. If a packet of status data has been received, process 67 checks the packet for invalid session id numbers, player id numbers, impossibly large movements of objects, and other signs of invalidity.

[0062] If no invalid data is found by process 67, process 68 processes the incoming status data records and updates the 3D variables representing player character motion and other changes in status indicated by the incoming data. Process 68 also checks the time stamp and message serial number to detect delayed data, to resolve conditions where one data record is received before a second record but with a later time stamp, and to resynchronize elapsed time with other systems.

[0063] Process 69 then updates the self view of the simulated 3D world and updates the image data from which a video signal is generated for display on the televison screen. One example of a change in status that can greatly affect the displayed image of the self player character is if another character has collided with the self character. The self video game system 42 discovers that such a collision has occurred by receiving a status data record from the system 42 that moved the colliding character into or near the coordinates of the self player character.

[0064] After the 3D data for the simulated game world has been updated by process 69, process 70 updates a 2D map of the simulated world and transmits the map in a compressed form that can be transmitted quickly to portable game systems 44 and 47 for display on LCD 22.

[0065] Process 72 checks for requests for an LCD display of another player's view of the simulated world (as described above with reference to Fig. 5) and generates data records containing the coordinates and orientation of all moving objects that can be viewed from the requested point of view and viewing angle. These generated data records are then transmitted through link 186 (Fig. 3 and 4) to portable game system 44 or 47 so that

processor 50 and coprocessor 301 can generate the requested image on LCD 22. Alternatively, process 64 may generate and transmit data for all moving objects to portable game system 44 or 47 at infrequent intervals whether requested or not.

[0066] Process 73 checks for requests for an LCD display of a map of a selected area of the simulated game world (as described above with reference to Fig. 6) and if requested, process 71 then generates map data records containing the coordinates and orientation of all moving objects that are represented within the selected map area. Process 71 then transmits these generated map data records through link 186 (Fig. 3 and 4) to portable game system 44 or 47 so that processor 50 can generate the requested map on LCD 22. Alternatively, process 71 may generate and transmit data for all moving objects to portable game system 44 or 47 at infrequent intervals whether requested or not.

[0067] Process 66 determines if the self status has changed, either because of the self player manually caused generation of control data that affected the self status or because an incoming status data record indicated that another character or object caused a change in self status. If process 66 determines that self status has changed, process 63 generates an outgoing status data record and transmits it to messaging server 120 to be shared with other video game systems 42.

[0068] Fig. 13 illustrates examples of several variable length record formats of status data records 78 that are transmitted through the Internet between video game systems 42. Each example record format has six fields in common: a record format code indicating which format a record has, an elapsed time stamp in milliseconds or other convenient time unit, a message serial number for each system, a session number assigned by the messaging server,

a player identifier that is different for each game system in a session, an object identifier that identifies a player character, non-player character, and moving objects such as a swinging door or falling rock. Non-moving object such as a wall have identifiers to identify collisions, but do not require location or movement data.

[0069] Movement records specify 3D cartesian spatial coordinates and velocity vectors. Orientation records specify heading (azimuth), pitch, and roll angles and rate of rotation of heading in case a character is turning around rapidly. Rates of rotation for the other two dimensions are not shown. Collision records specify which object collided with which object. Character animation records specify whether a character is swimming, flying, jumping, etc. Since these animations are stereotyped and preprogrammed, there is no need to specify every small motion. Attribute records specify attribute-value pairs such as degree of strength, type and number of weapons, weight of an object, rate of growth, and other attributes. Entry records specify which object is passing through which door or other portal and in which direction. Many other record types and formats may be used.

[0070] Fig. 14 illustrates two examples of variable length initializing record formats that are sent from messaging server 120 to each video game system 42 at the beginning of a session. The first record in Fig. 14 tells each system 42 which session id, game id, and player id to use in status records that each system 42 sends to the other systems 42. The Internet Protocol address field tells each video game system 42 what the current IP numbers are for each player, so that most message traffic can be sent direct through the Internet to all of the other systems 42, thereby reducting delays in transmission through messaging server 120.

[0071] The second record in Fig. 14 provides the screen name of each player to all of the other players in a session, so that word messages displayed on LCD 22 can refer to a specific player in systems that have a keyboard (see Fig. 9) for entering messages.

[0072] Fig. 15 illustrates use of multiple portable game systems 44 and 47 being used by player 10 as auxiliary displays, each displaying a different player character or other view of the simulated game world on an LCD 22. Multiple displays may be useful in war and crime games in which each LCD 22 is like a security camera or guard station monitor.

[0073] Fig. 16 is an example of players using conventional controllers 185 and 192 to control separate video game systems 42 that are linked to portable game systems 44 and 47 as substitutes for television screens. This embodiment may be useful in situations where a player does not have access to a television.

[0074] Fig. 17 illustrates a sequence of views of player characters displayed on an LCD 22 in a portable game system 44 or 47. A player can cycle through this sequence by pressing control members 14 or 15 on the portable game system.

[0075] Fig. 18 is a detailed block diagram of video game system 42 linked by a serial data communication link 45 to portable game system 44 or 47. Both systems 42 and 44 have CPU processors 86 and 50 and image co-processors 316 and 301 that generate simulated 3D images with rendered polygons or other methods of depicting 3D characters from different 3D points of view and viewing angles that may be rapidly changing. Characters generated by these four processors should have multiple body parts and joints that bend in a natural manner.

[0076] Software for use in video game system 42 and portable game system 44 or 47 is distributed on disk 43, which may be an optically readable disk such as a CD-ROM or DVD. Tracks 82 on disk 43 hold programs of executable instructions, graphics data for processing by those programs, and other kinds of data.

[0077] Programs accompanied by data are read from tracks 82 by disk reader 83 into RAM 90. These programs may be of two kinds: programs 151 (see Fig. 24) to be executed in cpu processor 86 and/or image co-processor 316, and programs 152 (see Fig. 24) to be downloaded through UART 88, serial data transmission link 45, UART 125 and stored in RAM 53 for execution in cpu processor 50 and/or image co-processor 301.

[0078] Optional external memory 16 may be a ROM cartridge, battery powered SRAM, or a data disk such as a CD or DVD. Programs accompanied by data, may be stored in external memory 16, read into RAM 53, and executed in processor 50 and co-processor 301. In the preferred embodiment, programs 152 and data are downloaded from RAM 90 in video game system 42 to RAM 53 in portable game system 44 and 47, so that all programs and data for a given game title may be distributed on disk 43.

[0079] Network interface 137 is activated by data from processor 86 which causes interface 137 to access messaging server 120 through the Internet 121 as described above with reference to Figures 1, 2, 13, and 14.

[0080] In Fig. 18, image coprocessor 301 may perform scrolling, flipping, blending, scaling, rotating, fading, windowing, coordinate transformations of polygons, texture rendering, bump mapping, lighting and shadows, rasterizing polygon data into displayable

pixel data in RAM (VRAM) 302 and other image processing functions for display on LCD 22.

[0081] Fig. 19 illustrates a time sequence, from top to bottom, of processing by two video game systems 42: (1) in the left column and video game system 42 (2) in the right column. Because of variable delays in transmission of packets through the Internet, some status data records 78 may be delayed more than others and some records may arrive out of sequence. This may cause an unintentional reversal of recent status data being replaced with older data. To prevent this problem, semaphores are used as illustrated in Fig. 19. A semaphore in this context, is data from one system to a second system that blocks further processing in the second system until the semaphore data is received by the second system.

[0082] In the Fig. 19 example, a first character controlled by system 1 is about to lock a door, but a second character controlled by system 2 is about to enter the same door. If the two characters arrive at the door at about the same time, one system may record the second character status as having entered, while the other system may record the second character status as having been blocked by the locked door. Later, if the second character is reported as beyond the door, the inconsistency in status can be automatically corrected. But it is better to avert this problem, because two player may be watching their first and second characters approach the door and should not be misled by false imagery. Whether the second character enters or is blocked, the status of the second character should be the same in all systems in a session.

[0083] In Fig. 19 at time 1, both systems agree on the "door is not locked" status. At time 2 system 1 sends a status record that notifies other systems of the intention to lock the door. At time 3 system 2 sends a status record that notifies other systems of the intention to enter the door. The two notices cross in cyberspace. At time 4 system 2 receives the lock notice record, but must disregard it until system 1 responds to the entry notice. At time 5 system 1 receives the entry notice. At time 6 system 1 sends a "go ahead" notice. At time 7 system 2 receives the "go ahead" notice. System 2 then updates the status of the second character as having passed through the door and updates the character's location as beyond the door. At time 8 system 2 sends this location change that confirms that the second character has passed through the door. At time 9 system 1 receives the location change record and updates the location status of the second character. At time 10 system 1 updates the status of the door as locked and sends a "door locked" notice. At time 11 system 2 receives the "door locked notice and updates the door status as locked. At time 11, both systems agree on the status of the locked door and the location of the second character.

[0084] To avoid a deadlock (actually livelock because it can be interrupted) situation in which each system is waiting for the other system to do something and thereby blocks further progress in the game in all systems, perhaps because one of the semaphore records failed to arrive or because of a bug in a program, there should be a time limit of about two seconds on semaphore records. After the time limit has passed, the location data should force a status change in all systems in a session so that the game can proceed.

[0085] If the above description of Fig. 19 seems like unnecessary complexity, remember that the goal is to allow all players to display what all characters are doing and to accurately display the results of their prior actions without transmitting huge amounts of video data

over the Internet. Transmitting video would not look real because of transmission delays and would not solve the problem of status updating.

[0086] Fig. 20 is a memory map of the digital programs and data stored on a computer-readable data storage disk product 43. Some of these programs are executed in video game system 42, some are executed in portable game system 44, and some are executed in both systems.

[0087] Fig. 21 is a typical memory map of programs and data stored in RAM 90 in each video game console 42. See Fig. 18 for RAM 90.

[0088] Fig. 22 is a three dimensional (3D) graph illustrating cartesian coordinates $(X_1\ Y_1\ Z_1)$ of an exemplary virtual camera 175 and coordinates $(X_2\ Y_2\ Z_2)$ of an exemplary object 171 being photographed by the virtual camera. See example in Fig. 8. Polar coordinates would also be an appropriate equivalent. For clarity, coordinates are not shown for camera 215 which may be the same as camera 175 but at different point of view location in the simulated 3D world. The viewing angle from camera 215 to object 172 is different than the viewing angle from camera 175 to object 172.

[0089] Fig. 23 is a perspective view of video game console 42 linked by data transmission link 45 to portable game system 44 that receives into socket 230 a memory cartridge 322 that also has a standard array of 12 button switches. This is to enable players who do not have a PC keyboard (as in Fig. 9) to enter phone numbers of Internet Service Providers (ISP) or Internet IP numbers to access messaging server 120. Instead, a player can enter digits of the phone number or IP numbers on the buttons switches in cartridge 322 which

also provides a ROM program to execute in processor 50 in portable game system 44 for displaying the prompting message "ENTER PHONE NUMBER" or "ENTER IP NUMBER" on LCD 22 and a program that transmits the entered number to video game system 42 through data link 45 or wireless equivalent.

[0090] Fig. 24 is a simplified block diagram illustrating downloading of game programs from video game system 42 to portable game system 44. When disk reader 83 reads game programs into RAM 90, the programs in this example are of two kinds, console program(s) 151 with associated data, and controller program(s) 152 with associated data. Controller program 152 is transmitted to RAM 53 in portable game system 44 and executed in microprocessor 50. Console program 151 is stored in RAM 90 and executed by microprocessor 86 which generates animated picture data 146 representing one or more animated characters with plural body parts performing an action. This picture data stored in RAM 146 is converted to a video signal by video generator 117 (see Fig. 18) that is passed to TV 11 by way of cable 41 (Fig. 18) and is displayed as animated pictures on TV screen 56. Microprocessor 86 also generates data records which it sends (arrow 154) to portable game system 44. Various record formats may be used by programs 151 and 152.

[0091] Fig. 24 illustrates data flow between two video game systems 42, messaging server 120, and portable game system 44. When the operators of the two systems 42 logon to messaging server 120, they are registered as clients on the messaging server. System 42 on the left loads a multiple-player game from disk 43 into RAM 90. Meanwhile System 42 on the right loads the same multiple-player game from disk 43 into RAM 90 on the right. Processors 86 in both systems 42 execute the game programs which generate data representing a simulated 3D game world in RAM 90 in both systems.

[0092] As the game progresses, system 42 on the left generates changes to the original game world that reflect movements of objects and other chages to variables in the simulated game world. These changes are are cumulative and in Fig. 25 are illustrated separately from the original game world. As described above with reference to Fig. 13, every time there is a change to a variable in the simulated game world, record is generated detailing the change. This record is transmitted through Internet 121 to system 42 on the right of Fig. 25, either directly or through messaging server 120. When system 42 on the right receives the change record, it is oprocessed and incorporated into the cumulative changes to the simulated game world in RAM 90 on the right so that both simulated game worlds including cumulative changes are substantially the same.

[0093] System 42 on the right generates first picture data for display of the motorcyclist character 18 moving in the simulated game world on TV 11 that agree with any changes to the game world that affect character 18 or other objects that he may be seen with.

[0094] System 42 on the right also generates and transmits game data to portable game system 44 through data link 45. This game data represents all of the changes that have occurred to the simulated game world that affect the area of the game world that will be displayed on LCD 22 on portable game system 44. In this example the operator of portable system 44 wants to display on LCD 22 the other player's character in the dinosaur scene that is also being displayed on TV screen 56.

[0095] The term "video" may include composite, non-composite, RGB, component, monochrome, color, analog, digital, raster scanned, and the like.

[0096] The term "portable game system" is a term of art that means a handheld game system that contains a discrete display device (e.g. LCD) and can be operated as an independent game system without any connection to other systems or displays.

[0097] The details of video game system 42 and portable game system 44 are given here only as examples and numerous other designs may be used.

[0098] The term "LCD" (liquid crystal display) has been used herein as an illustrative example of any display apparatus having discrete dot-matrix picture elements.

[0099] The term "program" as used herein may consist of more than one loadable module and typically includes executable instruction data and any other data that is typically part of a program module or modules.

[0100] Although I have described my invention with a degree of particularity in connection with what is presently considered to be the most practical and preferred embodiments, the foregoing description has been made only by way of illustration and example and is not to be interpreted as restrictive or limiting as to the meaning of words in the patent or its claims. It is understood that various modifications, variations, arrangements, and/or equivalents, can be devised without departing from the spirit and scope of the invention which is defined by the claims.

[0101] Reference Numbers in Drawings

10 human game player
11 television (TV) set or video monitor
12 human game player
14 control switch
15 direction control switch
16 external memory
17 player character
18 player character
19 linked system in general
20 joystick
22 LCD screen
33 LCD pictures
34 LCD pictures
40 serial data port in portable system
41 video signal cable to TV
42 video game system console
43 optical disk
44 portable game system
45 data link from console to portable game system
47 portable game system
49 cursor
50 cpu processor in portable system
53 RAM in portable system
56 video screen
57 control switch
60 program process
62 program process
63 program process
64 program process
65 program decision
66 program decision
67 program decision
68 program process
69 program process

70 program process
71 program process
72 program decision
73 program decision
76 boot ROM in portable system
78 data record
82 tracks molded into disk
83 optical disk reader
84 security processor
86 cpu processor in console system
87 serial data connector
88 serial port (UART)
90 RAM in console system
91 boot ROM in console system
92 address bus in portable system
93 data bus in portable system
117 video signal generator
119 LCD driver
120 messaging server
121 network
125 serial port (UART)
137 network interface
138 network interface
146 animated picture data
151 program for console system
152 program for portable system
153 data flow from portable to console
154 data flow from console to portable
157 program decision
158 program process
171 generic object
172 generic object
173 virtual camera
175 virtual camera
185 conventional handheld control unit
186 data link to auxiliary display

187 table or other physical support
192 conventional handheld control unit
215 virtual camera
228 generic object
229 player controlled object
230 cartridge socket
249 wireless data link
250 transceiver
251 transceiver
300 direct memory access (DMA)
301 image coprocessor
302 video RAM
316 image coprocessor
317 data bus in console
318 address bus in console
319 serial port (UART)
322 cartridge with button switches